MODIS
Version 1 Science Computing Facility
Software Delivery Guide

December 4, 1996

SDST-066
Change Notice 3

MODIS
Version 1 Science Computing Facility
Software Delivery Guide

Prepared by:

_____

James Firestone, SAIC/GSC — Date
SDST SSTG Manager

_____

Jean Wilkins, SAIC/GSC — Date
SDST Configuration Manager

Reviewed by:

_____

Laurie Schneider — Date
SDST R&QA Manager

_____

Frederick Patt, SAIC/GSC — Date
SDST System Analyst

_____

Barbara Putney, GSFC/Code 920.2 — Date
MODIS Systems Engineer

_____

Dr. Richard Isaacman, SAIC/GSC — Date
MODIS SDST Task Leader

Approved by:

_____

Edward J. Masuoka, GSFC/Code 920.2 — Date
MODIS SDST Manager

Change Record Page

This document is baselined and has been placed under Configuration Control. Any changes to this document will need the approval of the Configuration Control Board.

Document Title: MODIS Version 1 Science Computing Faciity
Software Delivery Guide

Document Date: April 30, 1996

Issue	Date	Page Affected	Description
Original	4/30/96	All	Baseline
Change Notice 1	7/24/96	3, 6, 8, 9, A-1 thru A-13	CCR #168
Change Notice 2	10/4/96	A-5, A-6	CCR #227
Change Notice 3	12/4/96	A-6	CCR #242

MODIS
Version 1 Science Computing Facility
Software Delivery Guide

Table of Contents

Table

MODIS
Version 1 Science Computing Facility
Software Delivery Guide

1. INTRODUCTION

This document describes considerations for science code development and delivery for the Version 1 (V1) Moderate Resolution Imaging Spectroradiometer (MODIS) production system. The document provides specifics relating to the contents of the delivery package submitted to the MODIS Configuration Management Officer (CMO), as well as discussion and pointers to relevant documents relating to Earth Science Data and Information System (ESDIS) and MODIS coding standards, V1 product specifications, ancillary and simulated datasets, and various other items required to make the science code production-ready.

1.1 Relevant Documents

Note: The latest version of each of the MODIS documents (i.e. the ones with Science Data Support Team (SDST) document numbers) can be found on the MODIS Programmers' Forum World Wide Web page at the Universal Resource Locator (URL) **http://modarch.gsfc.nasa.gov/programmers/Documents.html** or from the MODIS ftp site at: **ftp://ltpftp.gsfc.nasa.gov/pub/projects/modis/documents**.

- Data Production Software (DPS) and Science Computing Facility (SCF) Standards and Guidelines; EOSDIS 423-16-01; January 1994
- EOS Science Processing Database (for Product Identification); Ongoing Updates
- Interface Control Document Between the EOSDIS Core System (ECS) and Science Computing Facilities (SCFs); 209-CD-005-005
- MODIS-Application Program Interface User's Guide, Version 1.4; SDST-067; February 28, 1996
- MODIS Configuration Management Plan; SDST-004; June 30,1995
- MODIS Data Management Plan; SDST-006; October 25, 1995
- MODIS SDST Software Quality Assurance Plan; SDST-003; June 30, 1995
- MODIS Software Development Standards and Guidelines, Version 1; SDST-022A (CN 1); March 24, 1996
- MODIS Software Management Plan; SDST-002 (CN 2); March 24, 1996
- MODIS Science Data Processing Software Version 1 Requirements Specification; SDST-028; March 5, 1996
- MODIS Product Volumes and Process Load Estimates; SDST-009; August 11, 1995
- MODIS Version 1 Software Test Plan; SDST-060 (Review) ; May 1996
- Science Data Processing Segment (SDPS) Database Design and Database Schema Specification for the ECS Project, DID 311-CD-002-002

- Science Software Integration and Test Procedures for the MODIS Instrument at the GSFC DAAC; SDST-017A; March 29, 1996
- Science User's Guide and Operations Procedure Handbook for the EOS Core System (ECS) Project, Part 4; Software Developer's Guide to Preparation, Delivery, Integration, and Test with the ECS; 205-CD-002-002; Final, August 1995
- SDP Toolkit User's Guide for the ECS Project; DID 333-CD-003-002; August 1995
- Team Leader Working Agreement for MODIS Between EOS AM & PM Projects GSFC and the MODIS Science Team Leader; GSFC 421-12-14-02; April 21, 1994

2. VERSION 1 GOALS, REQUIREMENTS, AND CONFIGURATION MANAGEMENT

According to the Team Leader Working Agreement for MODIS Between Earth Observing System (EOS) AM & PM Projects at the National Aeronautics and Space Administration (NASA) Goddard Space Flight Center (GSFC) and the MODIS Science Team Leader, the V1 software is the first engineering version prior to the launch of EOS AM-1. Therefore, the software is expected to demonstrate all of its major functions, and send all status message outputs to production log files using standard error message interfaces. V1 is also, to the extent possible, supposed to provide a realistic assessment of Version 2 (V2) (i.e. launch-ready) computational resources.

In more specific terms, there are a number of goals and requirements associated with the V1 software development and delivery. The following are some excerpts from the MODIS Science Data Processing Software Version 1 Requirements Specification, SDST-028, with elaboration. Note that these are highlights only, and the full text of SDST-028 should be consulted for more information.

2.1 Version 1 Goals

Some of the overall goals for the V1 Release of the MODIS Science Data Processing Software (SDP S/W) from the Version 1 Requirements Specification are:

- "Demonstrate that each Science Team (ST) software developer has delivered working code that ingests simulated MODIS instrument data, or data derived from simulated MODIS instrument data" - note that a detailed schedule for simulated data development has been worked out after consultation with each of the MODIS discipline leads, so the SDST feels that it meets the anticipated needs of the V1 code. The simulated data set will include various scientific elements requested by the team. This schedule will be published on the Programmers' Forum home page as described in Section 1.1.
- "Demonstrate that working code which generates all at-launch products has been delivered" - that is, the code should employ science consistent with the latest version of the Algorithm Theoretical Basis Document (ATBD).
- "Employ the full set of ancillary data sets required by the V1 software processes. Use actual examples of the ancillary data as provided by the organization responsible for producing the ancillary data set. Ingest the ancillary data as it will be produced and interpolate or otherwise manipulate it to get the values at pixel locations as needed by the V1 algorithm" - strategies have already been developed by the team for dealing with gridded meteorological data from the National Oceanic and Atmospheric Administration (NOAA)/National Meteorological Center (NMC) and GSFC Data Assimilation Office (DAO). Many additional ancillary data capabilities will be available in the SDP Toolkit (SDPTK) 5.1 due out in May, 1996. In the meantime, if a Science Team Member (STM) has immediate needs for a particular dataset, they are asked to contact their Science Software Transfer Group (SSTG) representative in SDST for assistance.
- "Demonstrate that the V1 software generates appropriate Quality Assurance (QA) flags, warnings, operator messages and metadata and continues to process subsequent pixels when presented with defective inputs and/or radiances which are

inconsistent with anticipated values"- the flags and metadata should be included in the V1 Hierarchical Data Format (HDF) product specification as soon as they are known.

- "Baseline the HDF file specification for each MODIS at-launch product" - as with the simulated data, schedules have been worked out with each discipline for delivery of the V1 file specifications. The initial set is currently being baselined. When baselined, the file specifications will be held by the SDST CMO. Thereafter, all changes, deletions, and additions will be subject to review by the SDST Configuration Control Board (CCB), which meets weekly. The developer should contact the SSTG to arrange for completion of a Configuration Change Request (CCR).
- "Demonstrate use by MODIS science software of the HDF-EOS constructs and utilities" - at this time, the HDF-EOS library is under development. A prototype version is due out in the spring of 1996 with the release of SDPTK 5.1, and a full initial release in June, 1996. Therefore, due to the timing of the V1 MODIS deliveries, it appears likely that use of the HDF-EOS library by the developers will overall become a V2 goal. Time permitting, there may be a few cases where it is feasible to have SSTG perform initial tests with HDF-EOS on V1 code.
- "Identify the spatial and temporal dependencies between MODIS products and software processes needed to integrate the processes into the expected at-launch threads of execution and to provide the necessary Level 1 (L1) test data sets to test these dependencies" - the current dependencies are depicted in a set of "V1 bubble charts" (based initially on the dependencies seen in the Beta deliveries) and "At-launch bubble charts", both available on the Programmers' Forum home page. Each developer should check these charts for accuracy as soon as possible, and notify their SSTG contact with any changes. The goal for V1 is to have the V1 bubble charts and at-launch bubble charts match as closely as possible, given the current state of the science development for each product. These charts are important because they are used in constructing the threads for system and end-to-end tests by the SDST Test Team and ultimately the various DAACs. If there is a need to change the at-launch bubble charts, the changes should also be reflected in the next ATBD update performed by the STM. As for L1 test data sets, as described above a complete set of simulated datasets is being built by SDST for V1 testing.
- "Exercise the scheduling software chosen for the DAACs to verify that it meets MODIS's requirements and functions properly in the MODIS processing context" - ESDIS has chosen the **Autosys** software for this purpose, and SDST expects to exercise the software at the Team Leader Computing Facility (TLCF) during the V1 development period if it is available.
- "Test all V1 software and processing threads under both normal operation and exception-handling scenarios" - during V1, there will be a much more extensive set of test procedures developed than were in place for Beta. Tests will be included which will intentionally introduce errors into the processing stream (e.g. missing or corrupt files, anomalous data inputs, upstream processes which fail to complete, etc.). The SDST Test Team has already begun to define these procedures, and is seeking input and suggestions from the SCFs based on the tests they perform in developing the V1 code. Information of this type should be sent to Dr. Kenneth Mitchell, Test Lead for Levels 2, 3, 4 software, at mitchell@ltpmail.gsfc.nasa.gov. Please consult the V1

Requirements Specification, Section 3.4-4, for more detail on the characteristics of certain exceptions the V1 processes are expected to handle.

- "Provide accurate estimates of the computing resource requirements and output volumes of at-launch MODIS products based on actual software timing" - learning is ongoing in this area as well, including consultations with experts from Silicon Graphics, Inc. (SGI) and throughout the GSFC/EOS communities. Accurate estimates are particularly important because they are the basis for the procurement of the at-launch hardware which will run the MODIS and other EOS-AM production software. SDST will provide recommendations to the SCFs regarding specific strategies which can potentially improve code performance, and work with the SCFs to implement these strategies where it is mutually deemed appropriate.

2.1.1 Additional Goals Not Listed in the Version 1 Requirements Specification

- Ancillary data matched temporally to the time of the V1 simulated data set used in testing should be used - this is to ensure the scientific validity of the output product for the eventual at-launch algorithm, which will use actual L1B data. A time period of June 29, 1994 through July 1, 1994 will be used for unit and thread testing the V1 software.
- V1 code should provide error handling on the ancillary data sets read - i.e. if the ancillary data does not temporally or spatially match the simulated data, because of a file access error or it is not available, the code should write appropriate messages to the SMF log file and also provide for an alternate source of the data.
- Makefiles - a makefile template is still under development and will be released to the team shortly. All deliveries, before code acceptance is complete, should use this template in order to facilitate their maintenance, and to allow changes common to all deliveries (e.g. compilation options, directory locations for system-level libraries and files) to be made readily at the TLCF and DAACs.
- The software written in C language should not have any memory leaks, as determined by the Sentinel or similar commercial software product - memory leaks could cause unexpected outcomes in production and must be prevented. For SCFs not having access to memory leak checkers, SSTG, as part of the code acceptance process, will check incoming deliveries for memory leaks (preferably during one of the early iterations) and forward the output report to the SCF for further analysis. SSTG will make recommendations to the SCF regarding the cause or removal of memory leaks, but the SCF will bear primary responsibility for resolving memory leak-related issues.
- The delivery shall include a description of no more than twenty significant tests (e.g., limit tests) performed on the software at the SCF, including an accounting of the test data sets used and results of the tests. Knowledge of these tests will allow the SSTG and SDST Test Team to more effectively test the software at the TLCF and DAAC.
- Tolerances, binary information files (Science Software Integration and Test (SSI&T) requirements) - in Beta, MODIS was permitted to provide a single blanket tolerance value (2% was chosen) for the entire delivery to the GSFC DAAC, rather than individual tolerance files (each with ".tol" extension) for each array in the product HDF files. The tolerances are supposed to provide a measure of the allowable variability of

a particular data field attributable to the process of porting the software from the SCF platform to the DAAC's platform (currently an SGI Power Challenge system). Since, in the case of MODIS, the TLCF system where the delivered Beta product suite was created is the same Power Challenge as at the GSFC DAAC, no differences in particular data fields are expected (i.e. in the case of some other EOS-AM instrument teams, the SCFs deliver directly to the DAAC so the tolerance takes on more significance than for MODIS). No decision has yet been made as to whether tolerance files will be needed in V1, but SSTG will notify the SCFs for inputs and assist in generating the tolerance files if they are required.

Binary information files, as envisioned by the DAAC in Beta, were bit-level descriptions of each binary file written to a text file with a ".bif" extension. MODIS pointed out that this level of information was present in the Beta Processing Files Description document, so provision of binary information files would have been duplication. As with tolerances, no decision has yet been made as to whether separate binary information files will be needed in V1, but SSTG will work with the SCFs on providing these if they are required.

· Use of standard naming convention for makefiles, executables, file specifications, and PCFs as described in packing list contents in Appendix C.

2.2 Version 1 Requirements

Note: These items are summarized in the form of a checklist in Appendix F. The numbers indicated in parentheses after each quoted item are the requirement numbers taken from the V1 Requirements Specification document.

2.2.1 Interfaces

· "Each software process within the MODIS V1 SDP S/W system shall employ the Earth Science Data and Information System (ESDIS)-supplied SDPTK 5.1 mandatory functions to access the DAAC production environment and services. The mandatory functions include generic file Input/Output (I/O), metadata, error message transaction, process control, ancillary data access, spacecraft ephemeris and attitude, and time and date transformations (DID 333-CD-003-002, August 1995, Section 4.1)" (2.3-1) - the present schedule calls for SDPTK 5.1 to be released to MODIS on May 1, 1996. Therefore, since many algorithms are due for delivery prior to that date, SSTG will work with the SCFs to link in TK 5.1 and retest the delivery prior to the completion of code acceptance and final submission to the Configuration Management (CM). That is, the initial delivery to SDST will have to be using Toolkit Version 5 (TK5) rather than TK6. The primary new TK6 capabilities anticipated for SCF use are new ancillary data routines (e.g. to support DAO and NMC data residing in HDF-EOS forms) and the incorporation of HDF-EOS libraries. Because of the relatively late arrival of TK6 relative to the MODIS V1 deliveries, it is expected that most if not all of the new TK6 capabilities will be employed for the V2 rather than the V1 deliveries. Regarding metadata tools, these are already available in TK 5.1, and SSTG will assist the SCFs in utilizing these tools or the MODIS-Application Program Interface (M-API) to write EOSDIS Core System (ECS) and MODIS core and product-specific metadata to the output product files.

- "Each software developer, with the help of the SSTG, shall provide an HDF file description for the archivable MODIS product(s) produced by their software. This specification shall describe the structure and format of the HDF data objects in the files down to the bit level. This specification shall be provided to the SDST CMO no later than the delivery of the software to the SDST SSTG" (2.3-2) - this refers to the final version of the V1 specification. The initial draft is due at the time agreed to with SDST, and should contain as much core and product specific metadata as is known at that time. A draft MODIS Metadata Dictionary is found on the Programmers' Forum home page at:
  ftp://ltpftp.gsfc.nasa.gov/pub/projects/modis/metadata/metaMODLAND0103.html.

  The SSTG expects to iterate the file specifications with the SCFs until all V1 metadata, as well as QA flags if available, are included. The number of iterations should be minimized to the extent possible, since there are many downstream dependencies on the specifications. All changes will be subject to SDST CCB approval after the initial draft is submitted. SDST and SSTG strongly recommend that as much time a possible is spent up-front on design of the specifications to facilitate use by the other V1 processes.

- "The numbering scheme for the Process Control File (PCF) logical numbers used by the software processes to obtain data from the SDPTK shall follow a consistent convention to be defined by the SDST, and shall not conflict with the range of logical numbers reserved for use by the Toolkit routines (10,000 - 10,999)" (2.3-5) - for all subsequent development in V1 and V2, SDST has established the following logical number assignments which the developer should follow, in order to prevent logical number "collisions" (i.e. two processes using the same logical number for referencing different files) when thread testing begins:

Land:	200,000-299,999
Ocean:	300,000-399,999
Atmosphere:	400,000-499,999
Level 1A:	500,000-599,999
Geolocation:	600,000-699,999
Level 1B:	700,000-799,999

The SSTG lead in each of the land, ocean, and atmosphere disciplines will be responsible for coordinating amongst the various developers within their discipline, to assure that for a given process each is using a unique range of logical numbers within the overall range for the discipline. Even with the unique range assigned to each process, it will be necessary to pay particular attention to the assignments made for certain types of files: those read in by more than one process (e.g. ancillary data), and those written by one process and used by another process. Files falling into either of these categories will need to be "registered" by the SSTG discipline lead with the central Logical Unit Number (LUN) database maintained by SDST. Then processes accessing the file on either input or output use the same logical number to reference it. This ensures correct operation of the DAAC production system, and avoidance of unnecessary requests to the data server which loads the files. The discipline lead will notify the SCF with the appropriate logical number to use for a particular file in these categories, after consulting the database. This database will also be updated at least weekly on the MODIS ftp site at:

ftp://ltpftp.gsfc.nasa.gov/pub/projects/modis/delivery/tables/fileLUN.table.

Finally, although this is not a requirement, it is strongly recommended that the developer isolate the definition of logical numbers in each module doing I/O, either at the beginning of the code or in a centralized include file. This is preferable to hard-coded logical numbers within the body of the code, because it greatly facilitates maintenance in the event the numbers have to be changed in a subsequent delivery.

- "The numbering scheme for the Status Message Facility (SMF) seed numbering assignments used by the software processes to log error messages to the SDPTK shall follow a consistent convention to be defined by the SDST, within the range of values assigned by ECS to MODIS (35,000 - 39,999)" (2.3-6) - to meet this requirement, SDST has assigned the following discipline-specific ranges for SMF numbering:

Level 1A:	35000 to 35500
Level 1B:	36000 to 36500
Land:	37000 to 37500
Ocean:	38000 to 38500
Atmosphere:	39000 to 39500
Common:	39501 to 39999, includes: 39501 for Global SMF messages 39601 for Product Generation Executive (PGE) Script

As with the PCF numbering, the SSTG discipline leads will make the assignment of SMF ranges to each developer for particular processes. They will also determine commonalities amongst SMF messages provided by different developers and commonalities with messages provided by developers in other disciplines. In these cases, the message in question will be moved to the global SMF list, and the seed number reassigned to one from the global list. For V1, each process should be delivered with its own process-specific seed file, while the global seed file will be maintained by the SDST centrally for all processes.

2.2.2 Process Interface Requirements

- "Processes requiring HDF utilities to produce data files shall use HDF Version 4.0" (3.2-2) - by the end of the V1 period, both M-API (Version 2.0) and the HDF-EOS library will be built with HDF 4.0, so this requirement provides uniformity across the entire V1 delivery since certain processes may continue to use native HDF calls. Given the newness of the HDF 4.0 release, it is acceptable for the earlier deliveries to be built with HDF Version 3.3r4, with SSTG making the substitution to HDF 4.0 before final acceptance.
- "Each software process in the V1 release shall be capable of being ported to and running on an SGI platform under the IRIX 6.2 or higher operating system running in native (n32 or 64-bit)" (3.2-1) - note the acceptability of n32 mode; SSTG has found that 32-bit compilation using the "-n32" option provides similar performance to full 64-

bit operation. SDPTK 5.1, HDF 4.0, and M-API Version 2.0 will all be compiled in "-n32" mode.

2.2.3 Process Coding Standard Requirements

- "Each software process shall be coded according to the standards specified in the most recent baselined version of the MODIS Software Development Standards and Guidelines, SDST-022" (3.3-1) - this requirement refers to Version 1 of the MODIS Software Development Standards and Guidelines. The delivered code must include full ANSI compliance, provide for complete prologs, send all output to the SMF log file rather than the terminal, use the SDPTK for certain software functionality, and adhere to other standards specified by NASA ESDIS or MODIS SDST. Unless a follow-on waiver is specifically granted, all existing waivers granted during the Beta delivery DO NOT apply to the V1 delivery, i.e. the code will be expected to fully comply with the V1 standards. If a STM is unsure about the current status of a particular waiver, or has a new waiver in mind for the V1 code, they are urged to contact their SSTG discipline lead as soon as possible. In the case of new waivers, the SSTG lead will submit the request to ESDIS for review and keep the SCF updated on its status. The status of V1 waivers will be made available on the Programmers' Forum home page under "Current Events." During V1, the SSTG expects to make use of the new Programming Research/Quality Assurance (PR:QA) software tool at the TLCF, to automate the checking of incoming code for compliance with ESDIS and MODIS standards. Both C and FORTRAN versions of a PR:QA script to automate code checking have been developed by SSTG.

2.2.4 Product File and Specification Naming Conventions

- "The SDST shall define a consistent naming convention of the MODIS product files" (4.2-1) - the naming convention for V1 files is included as Appendix B. The convention for file specifications (e.g., HDF) convention shall be the same as the naming convention described in Appendix B, except it only includes the first five tokens (through level, and excluding temporal information and tile if applicable, etc.).

2.2.5 Metadata Requirements

The following explanation applies to requirements 4.3-1 to 4.3-4:

The draft MODIS Metadata Dictionary is found on the Programmers' Forum Web at: **ftp://ltpftp.gsfc.nasa.gov/pub/projects/modis/metadata/metaMODLAND0103.html**, and includes description of both ECS and MODIS core (i.e. present in all files) and land product-specific metadata. As mentioned previously, SDPTK 6.1 in conjunction with the M-API can be used to write the metadata, although the first submission of the V1 HDF file specifications do not need to include all of the metadata. SSTG will assist the SCFs in adding metadata to their output files and will place code templates and examples on the MODIS ftp site as they become available.

2.2.6 Quality Assurance Requirements

Regarding the QA requirements in Sections 4.5-1 and 4.5-2 of the V1 Requirements Specification, the flags should be present in the final version of the V1 product specifications delivered to the CM. They are not required with the initial draft of the product specifications.

2.2.7 Additional Items Needed from Science Software Deliveries

The following items are not currently requirements but have been submitted to the CCB for review and possible inclusion as Version 1 requirements.

- The final delivery to CM shall include a bit-level description of every data file the algorithm needs to run, including ASCII Look-up Tables (LUTs), binary files including ancillary data, and the output HDF product specification. The SSTG-provided template (Appendix E) shall be used for providing the information on ASCII and binary files; the product specifications may be provided in the text format used in Beta. The SSTG does expect to provide a common look and feel for the HDF product specifications prior to the completion of the baselined V1 Processing Files Description document for the DAAC, of which they are a part. The most likely scenario is that a software tool (e.g. National Center for Supercomputing Applications (NCSA) **hdp** or **Mosaic**) will be run on the final output product for each process, to provide a file specification with a uniform appearance and thus help automate the creation of the V1 Processing Files Description document.

The contents of Appendix E are available in both MS Word for Mac 6.0 or ASCII format, along with sample inputs, from the MODIS ftp site at:
ftp://ltpftp.gsfc.nasa.gov/pub/projects/modis/delivery/templates/filedesc.template.
If possible, file descriptions (except those for the HDF product specifications, whose delivery is subject to the agreed-to schedules) should be provided when known, in advance of the final SCF delivery. This is necessary because the V1 Processing Files is due in draft form in May, 1996. The MS Word for Mac 6.0 form of the templates are preferred with the delivery, because this software is used to produce the V1 Processing Files Description and other DAAC documents.

- The SCFs shall notify SSTG whenever major algorithm changes are implemented which are expected to increasing processing volumes and/or loads significantly. This allows the possibility of SSTG assistance in optimizing the code to the extent possible, well before the final delivery. For V1, SSTG plans, as part of the code acceptance process, to perform the computations of volumes and loads at the TLCF when a new delivery is received.

- The final delivery to CM shall include a PCF in the format of the template on the MODIS ftp site at:
  ftp://ltpftp.gsfc.nasa.gov/pub/projects/modis/delivery/templates/PCF.template.
  It is essential that no lines be removed from the template format to ensure compatibility at the DAAC.

2.3 Version 1 Configuration Management

One of the major lessons learned during the Beta software development was that tighter management and version control of HDF product specifications, software, and data sets is needed. Therefore, SDST expects that in V1 the SCFs will interact more frequently with the SDST CMO (modiscm@ltpmail.gsfc.nasa.gov). The CMO will be responsible for:

- maintaining the baselined set of HDF product specifications;
- making the baselined set available on the MODIS ftp site;
- notifying SCFs of changes to baselined HDF product specifications;
- checking in code deliveries from the SCFs, as well as SSTG after transfer activities are completed, and inspecting the deliveries for completeness;
- maintaining the MODIS ftp site, and the associated archive of documents, templates, datasets, tables, etc.
- maintaining a global SMF error message (seed) file used by all the disciplines in SDST.

When the V1 directory structures for the function libraries and test data sets are established, the SCFs will be notified by the CMO so they can retrieve these items as needed.

The process of code acceptance, as elucidated in Section 3, terminates with the CMO certifying that all of the required elements of a successful software delivery have been received. Until this certification takes place, the delivered process cannot be moved to the SDST Test Team for thread testing. Therefore, it is essential that the SCF work closely with SDST and their SSTG contact to account for all of the required elements. The CMO will maintain the deliveries under version control in UNIX Revision Control System (RCS). Once the handoff to the DAAC is complete, DAAC personnel will be responsible for managing the software in the Atria, Inc. ClearCase environment.

3. DEVELOPMENT APPROACH

The SSTG Code Acceptance Process describes the interactions between SSTG and the SCFs in developing the code, and the orderly sequence of events followed from the time the code is first received until its final delivery from SSTG to the SDST CMO (found on: **http://modarch.gsfc.nasa.gov/programmers/Procedures.html**). These events are designed to ensure that the code meets all ESDIS coding requirements and is ready for the production environment.

The Code Acceptance Process is the first of a three-part process between the initial delivery to the CMO and the final delivery from SSTG to CMO prior to thread testing. The second part involves peer review (a walkthrough) of changes made to the code by SSTG, to ensure that the changes were made correctly and completely. The walkthrough is an essential part of the SAIC Common Approach to Software Development, and follows procedures established by the Software Engineering Institute (SEI) at Carnegie Mellon University. It should also be noted that not only code is walked through, but design as well for utilities (e.g. L1B readers, spatial compositing routines, I/O interfaces) written by SSTG as part of their support for the SCFs. Part 3 involves SSTG's completion of the CM Transfer Checklist, which is a final verification that all of the items required in the code delivery and as output of the code acceptance process are accounted for. Note that the contents of the CM Transfer Checklist closely matches the content of the Packing List template found in Appendix C.

As with the Beta delivery, the SDST V1 development philosophy is to continuously iterate with each of the science algorithm developers. Once the SCF delivers software to the SDST CMO, the SSTG will typically perform a single, limited task on it and then return the updated software to the SCF. At that time, SSTG will discuss with the SCF the changes and mutually agree on the content and schedule for the next delta version. The next delta version could consist of an algorithm update (performed by STM) or a further SDST software update (performed by SDST). The turnaround time will normally be 10-15 working days, depending on the nature of the changes. These turnaround times will not apply to the first delivery and will reflect the schedules agreed upon as a result of recent meetings between SDST and the team. The schedule for turnaround will minimize the possibility of the algorithm diverging from the version that SSTG is modifying. The following are examples of limited tasks:

- Evaluating code for standards compliance.
- Restructuring module(s) to isolate I/O functions from the algorithm itself.
- Integrating SDPTK I/O or error handling routines.
- Integrating M-API.
- Adding core metadata to the output product.
- Optimizing specific portions of the code which are resource intensive.

3.1 Activities, Delivery Schedule, and Dependencies

The complete land, atmosphere and ocean team schedules are provided on the MODIS Programmers' Forum home page and will be updated every one to two weeks during V1.

4. THE DELIVERY PROCESS AND COMMUNICATING WITH THE SDST

Each STM is responsible for making his/her delivery to SDST electronically to a directory at the TLCF or on tape. Normally, electronic delivery is the most efficient method of transfer. However, tape may be preferable for major deliveries if large test datasets or other files are a part of the delivery. Copying the files electronically may be slow depending on the time of day, traffic volume on the local area network, and routing of the packets between the SCF and the TLCF. Otherwise, for incremental deliveries of source files, electronically will provide for faster turnaround. The SDST intends to return files to STMs by sending an e-mail message informing them to pick up the files from a directory on the TLCF. In V1, SDST intends to use a slightly different directory structure for SCF deliveries on the TLCF than that used in Beta (see Paragraph 4.1.2 for details). If there are additional deliveries which were not made in Beta (e.g. atmospheres Level 3), the SSTG will need to work with the SDST CM to determine an appropriate directory structure for each.

4.1 Delivery of Algorithm Package

The following are two types of deliveries for the algorithm package:

4.1.1 Tape Delivery

The SDST is currently able to accept 4mm (DAT), 8mm, DLT 2000 or 4000 series, and 9-track tape formats. If the SCF is making deliveries for multiple algorithms, the same tape may be used as long as the different deliveries are clearly marked. A separate tar file for each one is the preferred way of doing this. The packing list(s), in the form described in Appendix C, should be provided as files on the tape with the appropriate delivery as well as a hard copy with the tape. Those lists should indicate any special instructions for getting the delivery off the tape. The mailing address for tapes is:

General Sciences Corp. - MODIS Support Office
7501 Forbes Blvd., Suite 103
Seabrook, MD 20706
Attn: Configuration Management Officer

4.1.2 Electronic Delivery

This is the preferred method. There will be an account set up for each algorithm package for electronic transfer. The delivery structure is as follows: */delivery/process_ID,* where *process ID* is as specified in Appendix B. Each process ID will have two subdirectories, *INBOX* and *OUTBOX*. INBOX is for files incoming to the TLCF, and OUTBOX is for files outgoing from the TLCF. The protections on the directories will be set so that the SSTG programmer and the STM both have access to the appropriate directory. Ftp or kermit can be used for the transfer. The following steps can be used as a guide for making deliveries using ftp:

- Verify that you have ftp at your site. (It is standard on UNIX platforms.)
- Set your default to the directory where your files are at the SCF.

- Find out how to start ftp and attach to the modis-xl machine - the command will likely be “ftp modis-xl.gsfc.nasa.gov.”
- You will be asked to enter your modis-xl account name and password.
- Change directories to your INBOX by *cd /delivery/process_ID/INBOX*.
- Copy files from the SCF to modis-xl using the *put* command (e.g., *put abc.c* where abc.c is a file in your SCF directory). Use the *mput* command with wild cards (e.g., *mput *.c*) for multiple files.
- Verity that all the files you want to move to modis-xl are actually there.
- Enter *bye* or *quit* to exit ftp.

To pick up your files from the OUTBOX, a similar sequence of steps would be done except that once you had connected to modis-xl, you would *cd /delivery/process_ID/OUTBOX* and use the *get (or mget*)* command to copy files from modis-xl to the SCF. For larger deliveries, you may opt to use *tar* to compress the files before transferring to or from the TLCF. Also, please remember to remove the files in OUTBOX when they are successfully transferred to your SCF, preferably within a two-week period. This lets the CM know you have picked up the files, and also clears needed disk space on the TLCF.

4.2 Iterations with the SDST SSTG Staff

As in Beta, an SSTG programmer is assigned to work with one or more SCF algorithms. The SSTG programmer will review each algorithm, suggest changes, and make changes in coordination with the STM.

The following outlines the process for the science team and the SDST CMO for accepting the delivery code into SDST.

- The STM makes a delivery electronically to the correct delivery area or via tape.
- The STM sends mail to modiscm@ltpmail.gsfc.nasa.gov indicating that a delivery was made to a particular INBOX or that a tape has been mailed to the SDST.
- The CMO completes the checklist looking for particular items (e.g., build file or makefile, a single README file, and a single Packing List stored separately from the README file) in each delivery as summarized in Appendix G. Copies of the templates for the README file, Packing List, file descriptions, and PCFs are available from the MODIS ftp site at:
  ftp://ltpftp.gsfc.nasa.gov/pub/projects/modis/delivery/templates.

The CMO verifies the delivery against the checklist and if there are no omissions, sends e-mail to the STM indicating that the delivery was satisfactorily received. Otherwise, discrepancies are reported back to the STM and appropriate SSTG lead. The CMO and STM iterate this process until the delivery is complete.

When accepted the CMO will move the delivery to the software library for the SSTG programmer to work, and notification will be sent to them.

The following outlines the process for the STM and SSTG programmers to follow for making updates to the code. The SSTG recognizes that this process will vary from STM to STM with some STMs preferring to make all necessary code changes themselves and others preferring the SSTG staff to make the changes for them. More detail on all the required steps for code acceptance are at: **http://modarch.gsfc.nasa.gov/programmers/Procedures.html**. Note that in the following steps, STM may refer to the STM's designee:

1. Early in V1 development, the STM and SSTG programmer coordinate with each other to determine the division of labor between the two groups. These tasks should be small in scope to provide quick turnaround to the STM, so that the work of the STM and the SSTG programmer do not interfere. The programmer will also try to be cognizant of time when the STM will not be working on the algorithm, and make best use of that time. New/incremental deliveries by the STMs will follow steps outlined above. Changes made by the SSTG programmer will result in the following:

2. Documenting all changes in the prologue.

3. The SSTG programmer tests the program with the STM supplied tests to verify that he is still getting the same results, and the programmer uses the standards checker to verify he has followed all standards.

4. The SSTG programmer stores his version in the development library for tracking purposes and puts a copy in the */delivery/process_ID/OUTBOX* of the algorithm's delivery area.

5. The SSTG programmer sends an e-mail message to the STM to inform him/her to pick up the latest version of the code from that directory.

6. The STM copies the code to their home account, deletes it from OUTBOX, in a timely manner, does whatever verification he/she wants to and incorporates this code in the next delivery to the SSTG.

If the STM chooses not to accept a particular delivery from the SSTG programmer, the STM will review the reasons for that with the SSTG programmer. Some items such as the use of the SDPTK are not negotiable, other items such as optimization of the code are. To assist in code optimization efforts, SSTG can provide outputs of various profiling tools (e.g. *prof* and *pixie*) which analyze the execution patterns of the code and can be used to determine processing bottlenecks.

If the STM and the SSTG programmer can not resolve an issue between themselves, the SSTG programmer will notify the SSTG lead, who will work with the STM and the SSTG programmer to resolve it.

The SSTG will attempt to limit the number of different SSTG members with which a STM has to work; typically it will be a single individual but occasionally a second may offer assistance due to the number of algorithms delivered by some STMs. Initial assignments of SSTG programmers to STMs will occur internally to SSTG prior to the delivery, but they will be adjusted as the SSTG evaluates the deliveries that are received and workloads fluctuate.

The SSTG expects to fine-tune and adjust these procedures. Currently, many of the procedures and checklists are keyed to the requirements for delivery to the DAACs. As these requirements evolve, SDST plans to incorporate them into these SCF delivery instructions to minimize the amount of retooling needed once the software arrives at the TLCF. Suggestions on how to improve the process should be sent to the SDST Reliability and Quality Assurance (R&QA) Manager at lauries@ltpmail.gsfc.nasa.gov.

APPENDIX A: ACRONYMS

ATBD	Algorithm Theoretical Basis Document
CCB	Configuration Control Board
CCR	Configuration Change Request
CM	Configuration Management
CMO	Configuration Management Officer
DAAC	Distributed Active Archive Center
DAO	Data Assimilation Office
ECS	EOSDIS Core System
EOS	Earth Observing System
EOSDIS	Earth Observing System Data and Information System
ESDIS	Earth Science Data and Information System
GSFC	Goddard Space Flight Center
HDF	Hierarchical Data Format
I/O	Input/Output
L1	Level 1
LAI	Leaf Area Index
LUN	Logical Unit Number
LUT	Look-Up Table
M-API	MODIS-Application Program Interface
MODIS	Moderate Resolution Imaging Spectroradiometer
NASA	National Aeronautics and Space Administration
NCSA	National Center for Supercomputing Applications
NMC	National Meteorological Center
NOAA	National Oceanic and Atmospheric Administration
PCF	Process Control File
PGE	Product Generation Executive
PR:QA	Programming Research/Quality Assurance
QA	Quality Assurance
R&QA	Reliability and Quality Assurance
RCS	Revision Control System
SCF	Science Computing Facility
SDP	Science Data Processing
SDPS	Science Data Processing Segment

SDP S/W	Science Data Production Software
SDPTK	Science Data Production Toolkit
SDST	Science Data Support Team
SEI	Software Engineering Institute
SGI	Silicon Graphics, Inc.
SMF	Status Message Facility
SSI&T	Science Software Integration and Test
SSTG	Science Software Transfer Group
ST	Science Team
STM	Science Team Member
TK	Toolkit
TLCF	Team Leader Computing Facility
URL	Universal Resource Locator
V1	Version 1
V2	Version 2

APPENDIX B - MODIS VERSION 1 PRODUCT NAMING CONVENTION

Note: Within the naming conventions, bolded items are literally part of the name as indicated, while other parts of the name are defined as indicated.

B.1 Conventions for Level-1A, Level-1B, and Level-2 Products

MOD.AM1.V1.mnemonic**.**level**.D**yyyyddd**.**hhmmss.**D**YYYYDDD

where:

- **MOD** represents “MODIS,” and
- **AM1** represents the EOS-AM1 satellite,
- **V1** represents the SDPS Version 1 system of which this file is a part.
- mnemonic and level are as indicated in Table B-1 for each MODIS software process, and
- **D**yyyyddd represents the four-digit year followed by the Julian day within the year, for the first scan line of data within the file, and
- hhmmss represents the time of day applying to the first scan line within the file, in hours, minutes, and seconds, and
- **D**YYYYDDD represents the four-digit year followed by the Julian day within the year, for the file’s creation date.

B.2 Conventions for Level-2G and Level-3 Products

MOD.AM1.V1.mnemonic**.**level**.**[**D**yyyyddd1.]**D**yyyyddd2**.**[hhmmss2**.D**yyyyddd3**.**hhmmss3**.**] [subfile**.**][**h<**hnum>**.v<**vnum>**.r**].**D**YYYYDDD

where:

- **MOD** represents “MODIS,” and
- **AM1** represents the EOS-AM1 satellite, and
- **V1** represents the SDPS Version 1 system of which this file is a part.
- mnemonic and level are as indicated in Table B-1 for each MODIS software process (note that the Table does not yet include L3 atmosphere processes running in V1), and
- **D**yyyyddd1 represents the data day for the data within the file (Note: Used for MOD_PR18*, MOD_PR27_Y, MOD_PR28* files only), and
- **D**yyyyddd2 represents the four-digit year followed by the Julian day within the year, for the start date applying to the data within the file, and
- hhmmss2 represents the start time applying to data within the file, in hours, minutes, and seconds (Note: Used for MOD_PR18*, MOD_PR27_Y, MOD_PR28* files only), and

- **D**yyyyddd3 represents the four-digit year followed by the Julian day within the year, for the end date applying to the data within the file (Note: Used for MOD_PR18*, MOD_PR27_Y, MOD_PR28* files only), and
- hhmmss3 represents the end time applying to data within the file, in hours, minutes, and seconds (Note: Used for MOD_PR18*, MOD_PR27_Y, MOD_PR28* files only), and
- subfile is the subordinate file designator, and is one of [pcc, pnn, pxc] (Note: Used for MOD_PR27_Y file only), and
- hnum is the tile number in the horizontal direction for the global grid, and is zero-based (Note: Used for Land files only), and
- vnum is the tile number in the vertical direction for the global grid, and is zero-based (Note: Used for Land files only), and
- **D**YYYYDDD represents the four-digit year followed by the Julian day within the year, for the file's creation date.

B.3 File Naming Examples

a. Land example - the 16-day Level-3 BRDF product for December 31, 1998, at the 5th horizontal tile and 8th vertical tile, created on January 1, 1999, would be encoded as follows:

MOD.AM1.V1.brdf_16dy.L3.D1998365.h4.v7.D1999001

b. Oceans example - the L2 ocean color product suite for the data day December 31, 1998 and period Dec. 31, 1998 at 12:00 through Jan. 3, 1999 at 15:00 created on Jan. 1, 1999, would be encoded as follows:

MOD.AM1.V1.ocncolorlw_l2.L2.D1998365.D1998365.120000.D1999003.150000.D1999001

c. Atmospheres example - the L2 cloud mask product for Jan. 1, 1999 at 08:30, created on Jan. 2, 1999 would be encoded as follows:

MOD.AM1. V1.cldmask.L2. D1999001.083000. D1999002

d. L1B example - the L1B radiance product for Jan. 1, 1999 at 10:15:25, created on Jan. 2, 1999 would be encoded as follows:

MOD.AM1. V1.radiance.L1B. D1999001.101525. D1999002

Table B-1 File Name Summary by MODIS Process

Process ID	Product ID	Process Name	Level	Mnemonic
MOD_PRMGGA	MOD_PRMGGA	Tiled Geolocation Angular Data	L2G	geoang
MOD_PRMGPNTR	MOD_PRMGPNTR	L2G Pointer Map	L2G	pntr_1km, pntr_250m, pntr_500m
MOD_PRVOLC	N/A	Early Warning Volcano Alert	L2	volc
MOD_PR01	MOD01	Raw Counts	L1A	counts
MOD_PR02	MOD02	Calibrated Radiances	L1B	radiance
MOD_PR03	MOD03	Geolocation	L1A	geoloc
MOD_PR04.05.06A	MOD04/5/6	Aerosol joint product MOD04/5/6, daily	L3	atmos_040506_dy
MOD_PR04.05.06B	MOD04/5/6	Aerosol joint product MOD04/5/6,8-day	L3	atmos_040506_8dy
MOD_PR04.05.06C	MOD04/5/6	Aerosol joint product MOD04/5/6, Monthly	L3	atmos_040506_mn
MOD_PR04L	MOD04	Aerosol - Land	L2	aerland
MOD_PR04LA	MOD04A1	Aerosol Product, L3 Daily Interim	L3	int_aerland
MOD_PR04S	MOD04	Aerosol - Sea	L2	aersea
MOD_PR05	MOD05	Level 2 Near Infrared Precipitable Water	L2	pw_nir
MOD_PR06CT	MOD06	L2 Cloud Top Properties	L2	cld_top
MOD_PR06IR	MOD06	L2 IR Cloud Phases	L2	cld_IRphase
MOD_PR06OD	MOD06	L2 Cloud Optical Depth and Effective Radius	L2	cld_opt
MOD_PR07	MOD07, MOD08, MOD30, MOD38	Joint L3 Process to Generate Products 07, 08, 30, 38	L2	profiles
MOD_PR07A	MOD07A1, MOD08A1, MOD38A1	Joint L2 Process to Generate Products 07, 08, 38	L3	profiles
MOD_PR09	MOD09, MOD13	Surface Reflectance/Vegetation Indices / Thermal Anomaly	L2	srefl
MOD_PR09	MOD14	Surface Reflectance/Vegetation Indices / Thermal Anomaly	L2	fire
MOD_PR09G	MOD09G	Tiled Surface Reflectance	L2G	srefl_250M, srefl_500M
MOD_PR10	MOD10	Snow Cover	L2	snow
MOD_PR10A	MOD10A	Gridded Daily Snow Cover	L3	snow_dy
MOD_PR10G	MOD10G	Tiled Snow Cover	L2G	snow
MOD_PR11	MOD11G	Tiled Land Surface Temperature/ Emissivity	L2G	lst
MOD_PR11A	MOD11A	Gridded Land Surface Temperature/ Emissivity Preprocess	L2G	lst_pre
MOD_PR11B	MOD11B1	Gridded Land Surface Temperature/ Emissivity	L3	lst_8dy
MOD_PR11B	MOD11B2	Gridded Daily Land Surface Temperature/ Emissivity - Climate Modeling Grid	L3	lst_1dy_cmg
MOD_PR11C	MOD11C1, MOD11C2	Gridded Land Surface Temperature/ Emissivity - Climate Modeling Grid	L3	lst_8dy_cmg, lst_1m_cmg
MOD_PR12C	MOD12C1	Land Cover - Climate Modeling Grid	L3	lc_96dy_cmg
MOD_PR12M	MOD12M	Monthly Land Cover Type	L3	landcov_32dy
MOD_PR12Q	MOD12Q	Quarterly Land Cover Type	L3	landcov_96dy
MOD_PR13	MOD13	Gridded Vegetation Indices, 8-day	L3	vi_8dy

Process ID	Product ID	Process Name	Level	Mnemonic
MOD_PR13C	MOD13C	Gridded Vegetation Indices, 8-day - Climate Modeling Grid	L3	vi_8dy_cmg
MOD_PR13A	MOD13A	Gridded Vegetation Indices, 16-day	L3	vi_16dy
MOD_PR13AC	MOD13AC	Gridded Vegetation Indices, 16-day - Climate Modeling Grid	L3	vi_16dy_cmg
~~MOD_PR13A~~	~~MOD13P16~~	~~Gridded Vegetation Indices - 16-day, 250m, Pointer~~	~~L3~~	~~vi_pointer_16dy~~
~~MOD_PR13A~~	~~MOD13PC16~~	~~Gridded Vegetation Indices - 16-day, 250m, Pointer - Climate Modeling Grid~~	~~L3~~	~~vi_pointer_16dy_cmg~~
~~MOD_PR13A~~	~~MOD13R16~~	~~Gridded Vegetation Indices - 16-day, 250m, Reflectance~~	~~L3~~	~~vi_reflect_16dy~~
~~MOD_PR13A~~	~~MOD13RC16~~	~~Gridded Vegetation Indices - 16-day, 250m, Reflectance - Climate Modeling Grid~~	~~L3~~	~~vi_reflect_16dy_cmg~~
MOD_PR13B	MOD13B	Gridded Vegetation Indices - monthly	L3	vi_1m
MOD_PR13BC	MOD13BC	Gridded Vegetation Indices - monthly - Climate Modeling Grid	L3	vi_1m_cmg
~~MOD_PR13P~~	~~MOD13P8~~	~~Gridded Vegetation Indices - 8-day, 250m, Pointer~~	~~L3~~	~~vi_pointer_8dy~~
~~MOD_PR13P~~	~~MOD13R8~~	~~Gridded Vegetation Indices - 8-day, 250m, Reflectance~~	~~L3~~	~~vi_reflect_8dy~~
MOD_PR14A	MOD14A1, MOD14A2, MOD14A3, MOD14D	Multiday Composite Gridded Thermal Anomalies	L3	fire_8dy, fire_16dy, fire_1m
MOD_PR14C	MOD14C1, MOD14C2, MOD14C3	Multiday Composite Gridded Thermal Anamolies - Climate Modeling Grid	L3	fire_8dy_cmg, fire_16dy_cmg, fire_1m_cmg,
MOD_PR14G	MOD14G	Tiled Thermal Anamolies	L2G	fire_grid
MOD_PR15	MOD15	Leaf Area Index (LAI) and FPAR	L4	laifpar_8dy
MOD_PR15A1	MOD15A1	Daily Intermediate L3, LAI and FPAR	L3	laifpar_dy
MOD_PR15A2	MOD15A2	Daily LAI and FPAR	L4	laifpar_dy
MOD_PR15C	MOD15C1, MOD15C2	LAI and FPAR - Climate Modeling Grid	L4	laifpar_8dy_cmg, laifpar_1m_cmg
MOD_PR17	MOD17A1	Vegetation Production, Net Primary, 8-day	L4	psn_8dy
MOD_PR17	MOD17A2	Vegetation Production, Net Primary, yearly	L4	npp_yr
MOD_PR17	MOD17A3	GPP Intermediate	L4	gpp
MOD_PR17C	MOD17C1	Vegetation Production, Net Primary, 8-day - Climate Modeling Grid	L4	psn_8dy_cmg
MOD_PR17C	MOD17C2	Vegetation Production, Net Primary, yearly - Climate Modeling Grid	L4	npp_1yr_cmg
MOD_PR17P	MOD17P	DAO Climatology Compositing	L4	npp_scli
MOD_PR18_ DR1_3W	MOD19, MOD20, MOD23, MOD25, MOD26, MOD31	Ocean color derived 1, standard 3-week reference	L3	ocncolordr1_3wk
MOD_PR18_ DR1_D	MOD19, MOD20, MOD23, MOD25, MOD26, MOD31	Ocean color derived 1, standard daily	L3	ocncolordr1_dy
MOD_PR18_ DR1_I_3W	MOD19, MOD20, MOD23, MOD25, MOD26, MOD31	Ocean color derived 1, interim 3-week reference	L3	ocncolordr1_int_3wk

Process ID	Product ID	Process Name	Level	Mnemonic
MOD_PR18_ DR1_I_D	MOD19, MOD20, MOD23, MOD25, MOD26, MOD31	Ocean color derived 1, interim daily	L3	ocncolordr1_int_dy
MOD_PR18_ DR1_I_O	MOD19, MOD20, MOD23, MOD25, MOD26, MOD31	Ocean color derived 1, interim orbit	L3	ocncolordr1_int_orb
MOD_PR18_ DR1_I_W	MOD19, MOD20, MOD23, MOD25, MOD26, MOD31	Ocean color derived 1, interim weeky	L3	ocncolordr1_int_wk
MOD_PR18_ DR1_W	MOD19, MOD20, MOD23, MOD25, MOD26, MOD31	Ocean color derived 1, standard weekly	L3	ocncolordr1_wk
MOD_PR18_ DR2_3W	MOD21, MOD22, MOD24, MOD36, MOD39	Ocean color derived 2, standard 3-week reference	L3	ocncolordr2_3wk
MOD_PR18_ DR2_D	MOD21, MOD22, MOD24, MOD36, MOD39	Ocean color derived 2, standard daily	L3	ocncolordr2_dy
MOD_PR18_ DR2_I_3W	MOD21, MOD22, MOD24, MOD36, MOD39	Ocean color derived 2, interim 3-week reference	L3	ocncolordr2_int_3wk
MOD_PR18_ DR2_I_D	MOD21, MOD22, MOD24, MOD36, MOD39	Ocean color derived 2, interim daily	L3	ocncolordr2_int_dy
MOD_PR18_ DR2_I_O	MOD21, MOD22, MOD24, MOD36, MOD39	Ocean color derived 2, interim orbit	L3	ocncolordr2_int_orb
MOD_PR18_ DR2_I_W	MOD21, MOD22, MOD24, MOD36, MOD39	Ocean color derived 2, interim weekly	L3	ocncolordr2_int_wk
MOD_PR18_ DR2_W	MOD21, MOD22, MOD24, MOD36, MOD39	Ocean color derived 2, standard weekly	L3	ocncolordr2_wk
MOD_PR18_DR1_I_P	MOD19, MOD20, MOD23, MOD25, MOD26, MOD31	Ocean color derived 1, interim piece	L3	ocncolordr1_int_p
MOD_PR18_DR2_I_P	MOD21, MOD22, MOD24, MOD36, MOD39	Ocean color derived 2, interim piece	L3	ocncolordr2_int_p
MOD_PR18_L2	MOD18, MOD37	Ocean Color Lw, standard, L2 algorithms	L2	ocncolorlw_l2
MOD_PR18_L2	MOD19, MOD20, MOD23, MOD25, MOD26, MOD31	Ocean Color dr1, standard, L2 algorithms	L2	ocncolordr1_l2
MOD_PR18_L2	MOD21, MOD22, MOD24, MOD36, MOD39	Ocean Color dr2, standard, L2 algorithms	L2	ocncolordr2_l2
MOD_PR18_L2	MOD_PR18_QC	Ocean Color QC, interim, L2 QC	L2	ocncolorqc_l2
MOD_PR18_LW_3W	MOD18, MOD37	Ocean color Lw, standard 3-week reference	L3	ocncolorlw_3wk
MOD_PR18_LW_D	MOD18, MOD37	Ocean color Lw, standard daily	L3	ocncolorlw_dy
MOD_PR18_LW_I_3W	MOD18, MOD37	Ocean color Lw, interim 3-week reference	L3	ocncolorlw_int_3wk
MOD_PR18_LW_I_D	MOD18, MOD37	Ocean color Lw, interim daily	L3	ocncolorlw_int_dy
MOD_PR18_LW_I_O	MOD18, MOD37	Ocean color Lw, interim orbit	L3	ocncolorlw_int_orb
MOD_PR18_LW_I_P	MOD18, MOD37	Ocean color Lw, interim piece	L3	ocncolorlw_int_p

Process ID	Product ID	Process Name	Level	Mnemonic
MOD_PR18_LW_I_W	MOD18, MOD37	Ocean color Lw, interim weekly	L3	ocncolorlw_int_wk
MOD_PR18_LW_W	MOD18, MOD37	Ocean color Lw, standard weekly	L3	ocncolorlw_wk
MOD_PR27_Y	MOD27	Ocean productivity, L4 Yearly, Main file	L4	ocnprod_yr_m
MOD_PR27_Y	MOD27	Ocean productivity, L4 Yearly, Ppc Sub file	L4	ocnprod_yr_ppc
MOD_PR27_Y	MOD27	Ocean productivity, L4 Yearly, Pnn Sub file	L4	ocnprod_yr_pnn
MOD_PR27_Y	MOD27	Ocean productivity, L4 Yearly, Pxc Sub file	L4	ocnprod_yr_pxc
MOD_PR28D_I_O	MOD28	Sea surface temperature day, interim orbit	L3	sst_d_int_orb
MOD_PR28D_3W	MOD28	Sea surface temperature day, standard 3-week reference	L3	sst_d_3wk
MOD_PR28D_D	MOD28	Sea surface temperature day, standard daily	L3	sst_d_dy
MOD_PR28D_I_3W	MOD28	Sea surface temperature day, interim 3-week reference	L3	sst_d_int_3wk
MOD_PR28D_I_D	MOD28	Sea surface temperature day, interim daily	L3	sst_d_int_dy
MOD_PR28D_I_P	MOD28	Sea surface temperature day, interim piece	L3	sst_d_int_p
MOD_PR28D_I_W	MOD28	Sea surface temperature day, interim weekly	L3	sst_d_int_wk
MOD_PR28D_W	MOD28	Sea surface temperature day, standard weekly	L3	sst_d_wk
MOD_PR28N_3W	MOD28	Sea surface temperature night, standard 3-week reference	L3	sst_n_3wk
MOD_PR28N_D	MOD28	Sea surface temperature night, standard daily	L3	sst_n_dy
MOD_PR28N_I_3W	MOD28	Sea surface temperature night, interim 3-week reference	L3	sst_n_int_3wk
MOD_PR28N_I_D	MOD28	Sea surface temperature night, interim daily	L3	sst_n_int_dy
MOD_PR28N_I_O	MOD28	Sea surface temperature night, interim orbit	L3	sst_n_int_orb
MOD_PR28N_I_P	MOD28	Sea surface temperature night, interim piece	L3	sst_n_int_p
MOD_PR28N_I_W	MOD28	Sea surface temperature night, interim weekly	L3	sst_n_int_wk
MOD_PR28N_W	MOD28	Sea surface temperature night, standard weekly	L3	sst_n_wk
MOD_PR28_L2	MOD28	Sea surface temperature, Standard, L2 Algorithm	L2	sst_l2
MOD_PR28_L2	MOD28_QC	Sea surface temperature, Standard, L2 QC Info	L2	sstqc_l2
MOD_PR29	MOD29	Sea Ice Max Extent	L2	seaice_max
MOD_PR29A	MOD29A1	Gridded Daily Sea Ice Max Extent	L3	seaice_max_dy
MOD_PR29G	MOD29G	Gridded Sea Ice	L2G	seaice_max
MOD_PR32	MOD32	Calibration Data, BUOYMDB	L2	calib_buoymdb
MOD_PR32A	MOD32A1	Calibration Data, GLOBSUBS	L2	calib_globsubs
MOD_PR33	MOD33	Gridded Snow Cover	L3	snow_10dy
MOD_PR33C	MOD33C1, MOD33C2	Gridded Snow Cover - Climate Modeling Grid	L3	snow_10dy_cmg snow_1m_cmg

Process ID	Product ID	Process Name	Level	Mnemonic
MOD_PR35	MOD35	Cloud Mask	L2	cldmask
MOD_PR42	MOD42	Gridded Sea Ice Cover, 10-day	L3	seaice_10dy
MOD_PR42C	MOD42C1, MOD42C2	Gridded Sea Ice Cover - Climate Modeling Grid	L3	seaice_10dy_cmg, seaice_1m_cmg
MOD_PR43B1	MOD43A3	BRDF/Albedo (subsetting)	L3	brdfsubs
MOD_PR43B1	MOD43A4	BRDF/Albedo - Texture	L3	texture_16dy
MOD_PR43B2	MOD43A1	BRDF/Albedo	L3	brdf_16dy
MOD_PR43B2	MOD43A2	BRDF/Albedo	L3	bars_16dy
MOD_PR43BC	MOD43BC1, MOD43BC2	BRDF/Albedo - Climate Modeling Grid	L3	brdf_16dy_cmg, brdf_1m_cmg

APPENDIX C - PACKING LIST TEMPLATE

The following lists all of the items to be identified for the V1 packing lists. Please provide a single packing list with these items.

- The STM name, mailing address, and e-mail address to contact in case of problems.
- Name and description of all files in the delivery. For example:

File Name-Media ID	Format	Size	Description	Contact
ML1-soo.c	ascii	9872	main program for algorithm	Smith
ML1-abc.c	ascii	10511	subroutine to ML1-soo	Smith
ABC-test	HDF	999540	test data set	Smith

  - Separate input and output files for each process (including test data files and expected output from tests)
- Name of message text files (MODIS_#####.t)
- Complete names, brief descriptions, version numbers and TLCF locations of any files NOT included in the delivery, but associated with it (e.g. simulated datasets), along with a brief explanation of why not included
-
- Delivered PCF files using (Process ID).pcf naming convention. PCF structure follows verbatim the example on the MODIS ftp site (with no lines removed).
- Delivered makefiles using (Process ID).mk naming convention.
- Verify that there are no object files or executables in the delivery.
- Modify executable name in the Makefile to use convention (Process ID).exe.
-

APPENDIX D - README FILE CONTENTS

The following is a listing of all items identified in the V1 README files. Please provide a single README with all of these items.

- The STM name, mailing address, and e-mail address to contact in case of problems.
- Instruction on how to Build/Run the code.
- Name of environment variable in the Makefile used to point to include files.
- Which version of M-API was used . If M-API was not used, enter “No M-API used.”
- Which version of HDF was used.
- Which version of SDP Toolkit was used.
- Identify what vendor’s hardware (including model) and operating system (e.g., Silicon Graphics Indigo/IRIX6.1) were used to develop the code.
- Identify need for ancillary data sets.
- Identify what MODIS products are inputs, including channels used. This information should be consistent with the bubble chart update.
- Identify other types of input.
- Identify known software problems/defects (memory leaks; etc.).
- Total size of untarred delivery.
- Size of expected output files.
- Known differences between this software and at-launch version.
- Up to 20 tests performed at the SCF, including input/output data sets and test results.

APPENDIX E - FILE DESCRIPTION FORMAT TEMPLATE

The following is a sample template listing the items to be included in the Version 1 File Description Document:

File Format Description

File Name(s) (wildcard if appropriate):

MODIS Product:

File Type (ASCII, Binary):

File Source (Specify provider if applicable):

File Content Description:

File Structure Description:

Record/Line Format:

Parameter	Description	Start byte	Length

APPENDIX F - CHECKLIST FOR VERSION 1 CODING REQUIREMENTS

The following is a checklist for the Version 1 coding requirements:

1. All mandatory functions in SDP TK 5.1 (* includes metadata and ancillary data routines).

* 2. Notify SSTG when major algorithm changes are anticipated which could significantly increase volumes and loads.

* 3. Use PCF logical unit numbers falling within a range assigned to each discipline, and for the particular process. Processes reading ancillary data files or products written by other processes use exact logical unit numbers assigned to those files by SDST.

* 4. Use SMF seed numbers falling within a range assigned to each discipline, and for the particular process.

* 5. Use HDF 4.0, or M-API 2.0 when available.

6. Code adheres to all ESDIS/MODIS standards, subject to any waivers granted. New waivers to be requested through SSTG lead.

7. Consistent with standards, all messages are sent to SMF log files, none to the screen.

* 8. Use standard naming convention for product file, PCF, and makefile, as described in Appendix C. The standard naming convention for file specification is defined in 2.2.4.

* 9. Final V1 version of HDF output products include ECS core metadata with standard MODIS definitions, and product-specific metadata. Developer provides information needed for Metadata Configuration File.

* 10. Final V1 version of HDF output products include QA flags.

* 11. Ancillary data used by software is temporally matched to V1 L1B simulated data set and falls within time period established by SDST. Error handling on ancillary data sets.

* 12. Code uses SDST-developed makefile template when it is made available.

* 13. All memory leaks are resolved.

* 14. Description of all tests performed at the SCF in developing the code.

15. Physical delivery follows procedure established by CM for initial acceptance - copy to INBOX, notification sent to CM, all required elements provided.

Note: * = New for V1

APPENDIX G - CHECKLIST FOR ITEMS TO ACCOMPANY VERSION 1 DELIVERY

The following is a checklist for the items to accompany the Version 1 delivery:

* 1. HDF file specification draft delivered as per schedule and using the established naming convention described in 2.2.4 (prior to initial delivery to SDST). After baselining, any changes will be subject to SDST CCB review.

* 2. Deliver process-specific seed files (with initial delivery to SDST).

* 3. Delivery includes all required elements of a README file and packing list, as specified in this document. The README and packing list must each be delivered as a single ASCII text file separate from the other (with initial delivery to SDST).

4. Bit-level description of all data files, using SDST-developed template (prior to final delivery to SDST). These descriptions must be delivered as Microsoft Word for Mac Files, or files compatible with Microsoft Word for Mac.

* 5. Tolerances, binary information files if DAAC determines they are needed (with final delivery to SDST).

* 6. Data sets used as input to and output from significant SCF tests (with initial delivery to SDST).

Note: * = New for V1